E-Book

Why Real-Time Game Inventory Is Essential—and How to Deliver It

Part 3 of a 4 Part Gaming Series

redis

Introduction

Real-time inventory response is a crucial component of gaming. Any lags or freezes will lead to a disconnect between the player and the game, creating friction between the two. This is even more relevant given that today's games have evolved to become tremendously detailed, with large numbers of items and recipes in a player's inventory.

And all of those need to be accessed instantaneously to provide a seamless playing experience, which means processing millions of queries simultaneously. After all, having instant access to specific items in gameplay can be the difference between winning and losing.

To compound this even further, we live in an era where we expect responses to be delivered at lightning speed, or to be more precise, in real time. Any failure to satisfy these expectations frustrates players and hampers engagement, pushing them to games that can meet these demands.

This is especially relevant given the prominence of online gaming, where databases must provide local latency everywhere to avoid any lags in inventory lookups. Achieving this is by no means an easy feat, but with the right database, you can provide real-time inventory responses with local latency that are both reliable and scalable.

What does real time mean?

Applications have approximately 100 milliseconds (ms) before users feel like they're waiting for a response—one third of the time it takes to blink. To be considered real time, the game inventory request must be sent, processed, and then received in less than 100ms.

Expectations are higher than ever for game inventory responses

Inventories have become a crucial part of modern games. As gaming has progressed through the years, players expect games to have the capacity to accommodate a great number of items in their inventories and catalogs.

Gameplay mechanics can be enhanced by providing infinite inventory, but these come at a cost: lag times that get worse as the inventory builds up. Even with finite inventory games, there are different types of inventory to be accessed quickly. That's a lot of data to be processed in real time. Yet, despite this, meeting these demands has become fundamental to maximizing the playing experience.

Currency inventories

Currencies are the lifeblood of any in-game economy. They're used to make important transactions for items of value that pull players deeper into the game. Gold coins, purple diamonds, red orbs, you name it—if you want to gloss up that car with your favorite vinyl, you're going to have to fork out.

Just like in the real world, there's a desire for ownership of belongings. Players want to get their hands on the best inventory to sharpen their competitive prowess or even just to enjoy the thrills of character development.

The more inventory there is, the more there is to chase, hence games are now stacked full of inventories. To obtain these items, players need to generate currency. There are two types of currency in gaming:

- **Soft** – Earned via game play
- **Hard** – Earned by using physical money to buy virtual currency

Soft currency can be obtained by either playing the game or just waiting. Examples of this would be in Dynasty Warriors (players are rewarded with XP for completing battles), or coins in Idle Miner Tycoon (which are automatically given over time).

And then you have hard currency, which involves players spending real money to generate more virtual currency. Often this provides instant access to inventories as well as access to premium content (think of gems on Brawl Stars and gold bars in Candy Crush).

Having to accommodate both these types of currency proliferates the amount of inventories games have to process in real time. This is intensified even further when you take into account the number of players in a game and their currency requirements.

Player and character inventories

Character inventories refer to all of the items a character owns. In some games, players have multiple characters to build into teams. And as you'd expect, each of those characters will have an inventory of equipment, power ups, or anything else that makes up that inventory system.

But players also have their own inventory that can be distributed between their characters, creating a separate inventory system between the two. For example, in Fortnite a player can have an inventory of skins that they can equip their characters with. What's more is that an inventory can hold thousands of separate items, each with their own item count and current state.

When you add each player and their characters' inventories into the equation, databases end up with a colossal amount of data to process and update in real time. But here's another hurdle: characters are constantly shopping and changing their inventory as they progress deeper into the game.

Some games have a limited inventory system where characters are only allowed to carry a certain amount of inventory, forcing them to prioritize between items. To be able to juggle between different items and exchange old for new, the session stores behind the inventories need to respond in real time to create a seamless experience.

This is especially important for games that have extensive inventory systems. Escape from Tarkov, for example, allows players to store up to nearly 2,000 items in their inventory. The challenge this brings to the table is accessibility.

In many games, inventories are so vast that players have to dedicate time to sorting their catalogs. This is especially relevant for games similar to Escape from Tarkov, where their granular inventory systems make sorting almost mandatory.

And from a data perspective, these demands are astronomical. Games have millions of inventories and players are interacting with them all the time during gameplay, expecting responses to be in real time to meet their expectations.

Weapon inventories

First-person shooter (FPS) games like Call of Duty (COD) are fast-paced in nature and demand players react quickly and precisely to on-screen events. To put everything into context, the average FPS player has a reaction time of between 300 to 500 milliseconds, whereas for professional FPS players it's anywhere in the 100 to 250 millisecond range.

Now, remember our definition of real time? If a player needs to juggle between different inventory items, such as a pistol or an Mk47, then the whole process from request to processing to response to serving the players has to be completed within 100ms.

A slight delay in these data processing response times can be the difference between winning and losing, potentially robbing players of victory while also damaging the playing experience. And it won't take many losses due to lags before players start complaining, which can really hurt a game's community.

Across many games, being able to access and use inventory items at hyperspeed is a core aspect of gaming, making it almost a skill in itself. Real-time deficiencies in inventory responses will only frustrate players. In these scenarios, their inability to win might stem more from the deficiencies of the database rather than a lack of skill.

This creates a frustrating situation for the players due to being unable to carry out inventory commands in real time, hindering their ability to compete. Players have a low tolerance for jittery gameplay and so they'll simply jump to another game if frustrated.

Successful games require immersive and engaging experiences for players around the globe. Latency is the new outage and every millisecond counts. Success comes to those who offer performance and experience to encourage deeper, longer playing time.

The data technology requirements to pull off real-time game inventory responses

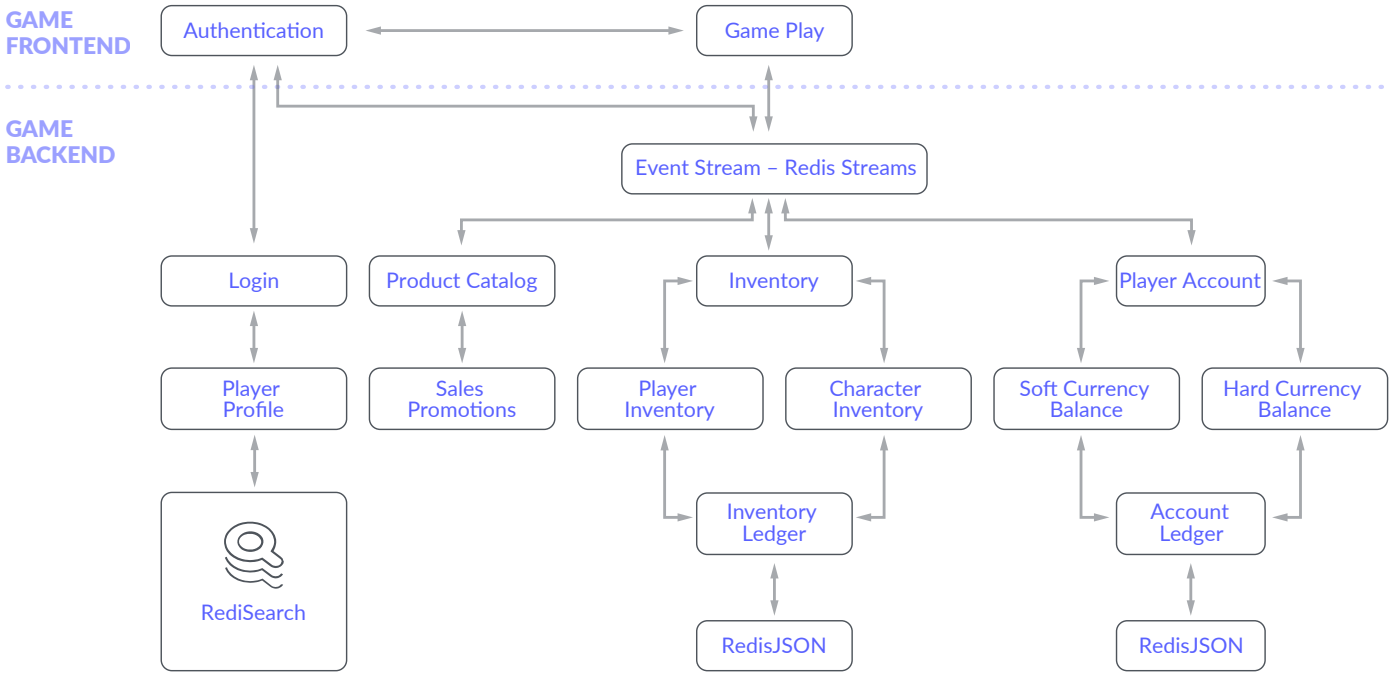
Achieving real-time game inventory response can be complicated. There could be millions of players who'll be requesting their unique account balances or querying for their unique player inventory. In such a case, developers need to integrate search into their database to cut down the number of transitions between services. Doing so will diminish the inventory response time as well as make the frontend more seamless.

Yet despite this, other factors come into play to generate inventory responses in real time. Data models must be compatible enough to provide tailored experiences for individual profiles through stored session states with unique IDs. Having numerous models of high availability and geographic distribution will give players local latencies.

Sorting and managing large streams of data is equally important. When events are created during gameplay or authentication, these events need to be allocated to the places where they'll be processed correctly.

To provide real-time inventory responses, a database needs to meet these technical requirements:

- **High database concurrency** – Databases will have potentially millions of players requesting their unique balances or flipping through inventories. Requests need to be processed simultaneously without any lags in performance.
- **Low latency** – Game inventory commands must be sent, processed, and then received in less than 100ms, requiring the database to be able to achieve <1ms latency to reach the 100ms goal.
- **Scalability with consistent performance** – The amount of inventory response requests can rise and drop in sudden surges. The database must be able to scale and deliver the same performance whether it's at the peak of the surge or the bottom of a trough.
- **Inventory response data model compatibility** – The right data models can bolster the speed of inventory responses, requiring any real-time inventory response database to include support and compatibility for those data models.
- **Integrated capabilities** – Incorporating search or other capabilities into the database will reduce the number of transitions between services, enabling the inventory response service to run faster as well as making the frontend experience more seamless.
- **Flexible deployability** – The inventory database needs to be deployed wherever the games are run, regardless of whether that's in the cloud, on-premises, or in a hybrid cloud environment.
- **Geo-distribution** - Reliability is just as crucial as speed. Millions of players across the globe will access their inventory and they expect to do so in real time, irrespective of their location.


GAME FRONTEND
GAME BACKEND
Authentication
Game Play
Event Stream – Redis Streams
Login
Product Catalog
Inventory
Player Account
Player Profile
Sales Promotions
Player Inventory
Character Inventory
Soft Currency Balance
Hard Currency Balance
RediSearch
Inventory Ledger
Account Ledger
RedisJSON
RedisJSON

Inventory responses need a real-time database

If your inventories are going to respond in real time, where players won't feel any latency, then the whole process that begins from request to response to processing the data for players has to be achieved in less than 100 ms.

A real hurdle is that data still has to travel to the servers, then be processed by the servers. All this can easily take 100ms, leaving somewhere between 0ms and 1ms for database latency. Unfortunately, traditional disk-based relational databases don't have the capabilities to process all this in such a short time frame.

To achieve this at such a high speed, you'll need to equip yourself with an in-memory NoSQL database—and that's where a database like Redis comes in. The most loved database in the world, Redis is an in-memory NoSQL database that can carry out 50 million operations per second at <1m latency.

The database must also include features like native support for data models for inventory responses. Redis Enterprise does this through a combination of enterprise-grade architecture and modules. The Redis Streams module can improve frontend functionality, while the RediSearch module provides search and analytics capability. And RedisJSON can bolster performance by storing data in a dynamic schema and retrieving documents at lightning speed.

But with potentially millions of players from around the world looking to access their player and character inventory at the same time, speed isn't enough. A complete real-time database needs to provide real-time responsiveness at any scale, with high availability on a global level. And that demands geo-distribution, automated resharding, single-digit automated failover, 99.999% availability, and the ability to be deployed in any environment.

Conclusion

Real-time inventory response is crucial in modern day games. Having a real-time database makes it easier to maintain player engagement, which in turn increases playing time, DAUs, MAUs, and, of course, revenue.

To learn more, visit redis.com/gaming or download the *Level Up Your Gametech with a Real-Time Database* white paper.

About Redis

Data is the lifeline of every business, and Redis helps organizations reimagine how fast they can process, analyze, make predictions, and take action on the data they generate. Redis provides a competitive edge to any business by delivering open source and enterprise-grade data platforms to power applications that drive real-time experiences at any scale. Developers rely on Redis to build performance, scalability, reliability, and security into their applications.

Born in the cloud-native era, Redis uniquely enables users to unify data across multi-cloud, hybrid and global applications to maximize business potential. Learn more about Redis at redis.com and sign up for your free trial.